

# Linux服务器构建与运维管理

## 第6章：文件服务器

阮晓龙

13938213680 / rxl@hactcm.edu.cn

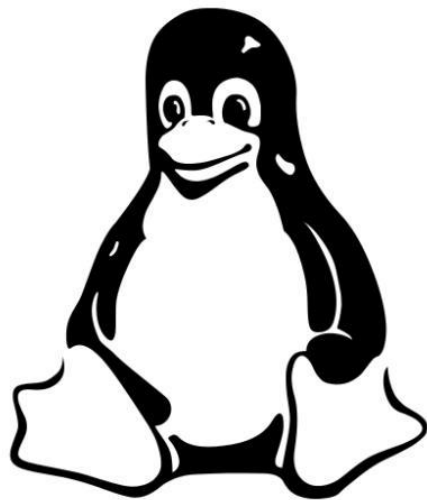
<http://linux.xg.hactcm.edu.cn>

河南中医药大学信息管理与信息系统教研室  
信息技术学院网络与信息系统科研工作室

2020.5

# 提纲

- FTP服务器
  - 基本原理
  - 使用vsftpd构建FTP服务
  - 实例：企业内部FTP文件服务
- NFS服务器
  - 基本原理
  - 构建NFS文件服务
  - 实例：工作组内的网络共享存储服务
- Samba服务器
  - 基本原理
  - 构建Samba服务
  - 实例：构建面向全终端的文件共享服务



# 1.FTP服务器

## 1.1 FTP的基本原理

- FTP是文件传输协议 (File Transfer Protocol)
  - 属于TCP/IP协议簇的一部分
  - 工作于OSI七层模型的应用层、表示层和会话层
  - 控制端口号为21，数据通信端口号为20
  
- FTP用于控制文件的双向传输，是Internet文件传送的基础，目标是提高文件的共享性，提供非直接使用远程计算机，使存储介质对用户透明和可靠高效地传送数据。
  - FTP支持跨路由的通信，能够在全互联网上提供服务



# 1.FTP服务器

## 1.1 FTP的基本原理

- 使用FTP服务需要拥有该FTP服务器授权的用户标识和口令进行登录，在远程主机上获得相应的权限后，才可以使用FTP服务器提供的服务。
- 在互联网中有一部分FTP服务属于“匿名（anonymous）”的，即匿名FTP服务器。
  - 匿名FTP服务器的目的是为公众提供文件下载服务，不要求用户必须是该FTP服务器的登记注册用户。
  - 匿名FTP服务器访问时也是有用户名的。
    - 用户名是特殊用户名：anonymous



# 1.FTP服务器

## 1.1 FTP的基本原理

- FTP支持多种文件传输方式，这些格式通常由FTP系统决定。
  - 文本方式：
    - 在文本传输模式中，其传输方式会进行调整，主要体现为对不同操作系统的回车、换行、结束符等进行转译，将其自动文件转译成目的主机的文件格式。
  - 二进制方式：
    - 在二进制传输中，保存文件的位序，以便原始和拷贝是逐位对应的，该传输方式不对文件做任何修改。



# 1.FTP服务器

## 1.1 FTP的基本原理

- FTP的工作模式分为两种：
  - 模式1: Standard (也为PORT, 主动模式)
    - FTP的客户端发送PORT命令到FTP服务器进行端口确认。
  - 模式2: Passive (也为PASV, 被动模式)
    - FTP的客户端发送PASV命令到FTP服务器进行端口确认。



# 1.FTP服务器

## 1.1 FTP的基本原理

### □ FTP：Standard模式

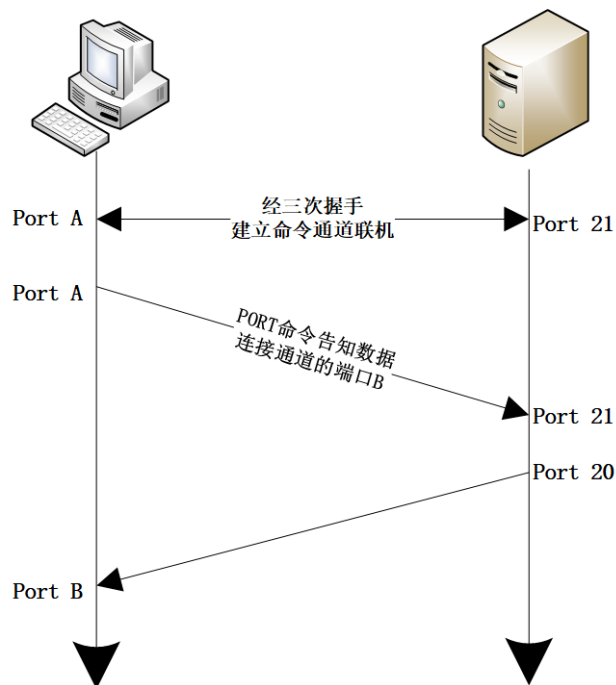
- FTP客户端首先需要和FTP服务器的TCP 21端口建立连接，通过这个通道客户端发送用户名和密码进行登录，登录成功后要展示文件清单列表或者读取数据时，客户端随机开放一个临时端口（又名自由端口，端口号在1024至65535之间），发送PORT命令到FTP服务器，“告诉”服务器，客户端采用主动模式并开放端口。
- FTP服务器收到PORT主动模式命令和端口号后，服务器的TCP 20端口和客户端开放的端口连接。
- 在主动模式下，FTP服务器和客户端必须建立一个新的连接进行数据传输。



# 1.FTP服务器

## 1.1 FTP的基本原理

### □ FTP：Standard模式



# 1.FTP服务器

## 1.1 FTP的基本原理

### □ FTP：Passive模式

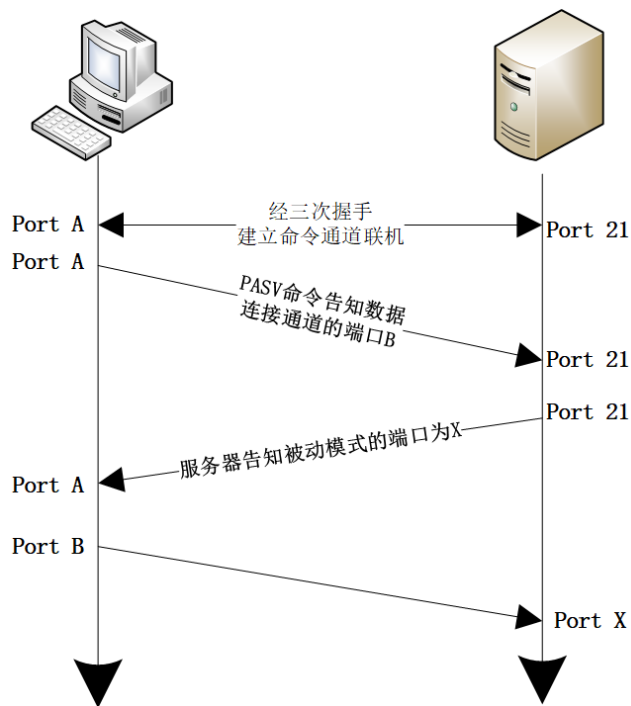
- FTP客户端连接到FTP服务器的TCP 21端口，发送用户名和密码进行登录，登录成功后要展示文件清单列表或者读取数据时，发送PASV命令到FTP服务器。
- 服务器在本地随机开放一个临时端口，然后把开放的端口告诉客户端，客户端再连接到服务器开放的端口进行数据传输。
- 在被动模式下，不再需要建立一个新的FTP服务器和客户端的连接。



# 1.FTP服务器

## 1.1 FTP的基本原理

### □ FTP: Passive模式



# 1.FTP服务器

## 1.1 FTP的基本原理

- FTP：Standard模式与Passive模式的区别
  - 主动模式和被动模式的区别可概述为两个方面：
    - 主动模式传输数据时是服务器连接到客户端的端口，被动模式传输数据是客户端连接到服务器的端口。
    - 主动模式需要客户端必须开放端口给服务器，很多客户端都是在防火墙内，开放端口给FTP服务器访问比较困难，被动模式只需要服务器端开放端口给客户端连接即可。



# 1.FTP服务器

## 1.1 FTP的基本原理

- 根据使用者的登录情况，FTP服务器的账号可分为实体用户、访客、匿名用户三种。
  - 实体用户（Real User）
    - FTP服务器默认允许实体用户（即系统用户）的登录。
    - 以实体用户做为FTP服务器的身份登录时，系统默认不对实体用户进行任何限制，该用户可以针对整个文件系统进行自身权限的工作，即FTP服务器的管理员权限。
  - 访客身份（Guest）
    - 在使用FTP服务器时，往往会给不同的部门或者某个特定的用户设置一个专属的帐户，创建一个访客身份就可满足。
  - 匿名身份（Anonymous）
    - 匿名用户即不需通过账户密码就可登录访问FTP服务器资源的用户，这类用户在FTP服务器中没有确切的指定账户，但可以访问FTP服务器中开放的文件资源。
    - 需要FTP服务器端允许匿名用户访问。



# 1.FTP服务器

## 1.1 FTP的基本原理

□ FTP属于Client/Server (C/S) 结构，包含客户端和服务端两部分。

■ FTP客户端程序：

- fileZilla Client
- FireFTP
- NcFTP



■ FTP服务端程序：

- WU-FTPD
- ProFTPD
- vsftpd



# 1.FTP服务器

## 【功能】

ftp 命令是用来控制在本地机和远程机之间传送文件，该命令可用于客户端与连接的 FTP 服务器间进行文件上传与下载。

## 【语法】

```
ftp [选项] [参数]
```

## 【选项说明】

ftp 命令选项及说明，如表 6-1 所示。

表 6-1 ftp 命令选项说明

| 选项 | 说明                                                     |
|----|--------------------------------------------------------|
| -d | 启用调试，显示所有客户端与服务器端传递的命令                                 |
| -v | 禁止显示远程服务器相应信息                                          |
| -n | 禁止自动登录                                                 |
| -i | 多文件传输过程中关闭交互提示                                         |
| -g | 禁用文件名通配符，允许在本地文件和路径名中使用                                |
| -s | 指定包含 FTP 命令的文本文件；命令在 FTP 启动后自动运行。此参数中没有空格。可替代重定向符（>）使用 |
| -a | 在绑定数据连接时使用所有本地接口                                       |
| -w | 覆盖默认的传输缓冲区大小 65535                                     |

## 【参数说明】

ftp 命令参数及说明，如表 6-2 所示。

表 6-2 ftp 命令参数说明

| 参数 | 说明                        |
|----|---------------------------|
| 主机 | 指定要连接的 FTP 服务器的主机名或 ip 地址 |

## 1.2 FTP文件传输命令

命令行

FTP

客户端软件



表 6-3 ftp 操作命令

| 命令                               | 说明                                                     |
|----------------------------------|--------------------------------------------------------|
| ! <code>[ shell [srg] ]</code>   | 在本地机中执行交互 shell, exit 回到 ftp 环境, 如: !ls*、zip           |
| macro-ame[ <code>args</code> ]   | 执行宏定义 macro-name                                       |
| account[ <code>password</code> ] | 提供登录远程系统成功后访问系统资源所需的补充口令                               |
| append local-file[remote-file]   | 将本地文件追加到远程系统主机, 若未指定远程系统文件名, 则使用本地文件名                  |
| ascii                            | 使用 ascii 类型传输方式                                        |
| bell                             | 每个命令执行完毕后计算机发出一声提示音                                    |
| bin                              | 使用二进制文件传输方式                                            |
| bye                              | 退出 ftp 会话过程                                            |
| case                             | 在使用 mget 时, 将远程主机文件名中的大写转为小写字母                         |
| cd remote-dir                    | 进入远程主机目录                                               |
| cdup                             | 进入远程主机目录的父目录                                           |
| chmod mode file-name             | 将远程主机文件 file-name 的存取方式设置为 mode, 如: chmod 777 a、out    |
| close                            | 中断与远程服务器的 ftp 会话 (与 open 对应)                           |
| cr                               | 使用 ascii 方式传输文件时, 将回车换行转换为回行                           |
| delete remote-file               | 删除远程主机文件                                               |
| debug[debug-value]               | 设置调试方式, 显示发送至远程主机的每条命令, 如: deb up 3, 若设为 0, 表示取消 debug |
| dir[remote-dir][local-file]      | 显示远程主机目录, 并将结果存入本地文件                                   |
| form format                      | 将文件传输方式设置为 format, 缺省为 file 方式                         |
| get remote-file[local-file]      | 将远程主机的文件 remote-file 传至本地硬盘的本地文件                       |
| glob                             | 设置 mdelete, mget, mput 的文件名扩展, 缺省时不扩展文件名, 同命令行的-g 参数   |
| hash                             | 每传输 1024 字节, 显示一个 hash 符号 (#)                          |
| help[ <code>cmd</code> ]         | 显示 ftp 内部命令 cmd 的帮助信息, 如: help get                     |

# 1.FTP服务器

## 1.2 FTP文件传输命令

命令行  
FTP  
客户端软件



ftp 常用操作命令的选项及其说明如表 6-3 所示。

表 6-3 ftp 操作命令

| 命令                            | 说明                                                                                                                         |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| idle[seconds]                 | 将远程服务器的休眠计时器设为[seconds]秒                                                                                                   |
| image                         | 设置二进制传输方式（同 bin）                                                                                                           |
| lcd[dir]                      | 将本地工作目录切换至 dir                                                                                                             |
| ls[remote-dir][local-file]    | 显示远程目录 remote-dir，并存入本地文件 local-file                                                                                       |
| macdef macro-name             | 定义一个宏，遇到 macdef 下的空行时，宏定义结束                                                                                                |
| mdelete[remote-file]          | 删除远程主机文件                                                                                                                   |
| mdir remote-files local-file  | 与 dir 类似，但可指定多个远程文件，如：mdir *、o、*、zipoutfile                                                                                |
| mget remote-files             | 传输多个远程文件                                                                                                                   |
| mkdir dir-name                | 在远程主机中建一目录                                                                                                                 |
| mls remote-file local-file    | 同 nlist，但可指定多个文件名                                                                                                          |
| mode[modename]                | 将文件传输方式设置为 modename，缺省为 stream 方式                                                                                          |
| modtime file-name             | 显示远程主机文件的最后修改时间                                                                                                            |
| mput local-file               | 将多个文件传输至远程主机                                                                                                               |
| newer file-name               | 如果远程机中 file-name 的修改时间比本地硬盘同名文件的时间更近，则重传该文件                                                                                |
| nlist[remote-dir][local-file] | 显示远程主机目录的文件清单，并存入本地硬盘的 local-file                                                                                          |
| nmap[inpattern outpattern]    | 设置文件名映射机制，使得文件传输时，文件中的某些字符相互转换，如：nmap \$1、\$2、\$3[\$1、\$2]、[\$2、\$3]，则传输文件 a1、a2、a3 时，文件名变为 a1、a2。该命令特别适用于远程主机为非 UNIX 机的情况 |
| ntrans[inchars[outchars>]     | 设置文件名字符的翻译机制，如 ntrans1R，则文件名 LLL 将变为 RRR                                                                                   |
| open host[port]               | 建立指定 ftp 服务器连接，可指定连接端口                                                                                                     |
| passive                       | 进入被动传输方式                                                                                                                   |
| prompt                        | 设置多个文件传输时的交互提示                                                                                                             |
| proxy ftp-cmd                 | 在次要控制连接中，执行一条 ftp 命令，该命令允许连接两个 ftp 服务器，以在两个服务器间传输文件。第一条 ftp 命令必须为 open，以首先建立两个服务器间的连接                                      |

# 1.FTP服务器

## 1.2 FTP文件传输命令

命令行  
FTP  
客户端软件



ftp 常用操作命令的选项及其说明如表 6-3 所示。

表 6-3 ftp 操作命令

| 命令                                | 说明                                                        |
|-----------------------------------|-----------------------------------------------------------|
| put local-file[remote-file]       | 将本地文件 local-file 传送至远程主机                                  |
| pwd                               | 显示远程机的当前工作目录                                              |
| quote arg1, arg2...               | 将参数逐字发至远程 ftp 服务器, 如: quote syst                          |
| reget remote-file[local-file]     | 类似于 get, 但若 local-file 存在, 则从上次传输中断处续传                    |
| rhel[cmd-name]                    | 请求获得远程主机的帮助                                               |
| rstatus[file-name]                | 若未指定文件名, 则显示远程机的状态, 否则显示文件状态                              |
| rename[from][to]                  | 更改远程主机文件名                                                 |
| reset                             | 清除回答队列                                                    |
| restart marker                    | 从指定的标志 marker 处, 重新开始 get 或 put, 如: restart 130           |
| rmdir dir-name                    | 删除远程主机目录                                                  |
| runique                           | 设置文件名只一性存储, 若文件存在, 则在原文件后加后缀 1, 2 等                       |
| sendport                          | 设置 PORT 命令的使用                                             |
| site arg1, arg2...                | 将参数作为 SITE 命令逐字发送至远程 ftp 主机                               |
| size file-name                    | 显示远程主机文件大小, 如: site idle 7200                             |
| status                            | 显示当前 ftp 状态                                               |
| struct[struct-name]               | 将文件传输结构设置为 struct-name, 缺省时使用 stream 结构                   |
| sunique                           | 将远程主机文件名存储设置为只一 (与 runique 对应)                            |
| system                            | 显示远程主机的操作系统类型                                             |
| tenex                             | 将文件传输类型设置为 TENEX 机的所需的类型                                  |
| tick                              | 设置传输时的字节计数器                                               |
| trace                             | 设置包跟踪                                                     |
| type[type-name]                   | 设置文件传输类型为 type-name, 缺省为 ascii, 如: type binary, 设置二进制传输方式 |
| umask[newmask]                    | 将远程服务器的缺省 umask 设置为 newmask, 如: umask 3                   |
| user user-name[password][account] | 向远程主机表明自己的身份, 需要口令时, 必须输入口令, 如: user anonymous my@email   |

# 1.FTP服务器

## 1.2 FTP文件传输命令

命令行  
FTP  
客户端软件



# 1.FTP服务器

## 1.3使用vsftpd构建FTP服务

- vsftpd是Linux操作系统中最常用的FTP服务器程序。
- vsftpd在安全性、高性能及稳定性三个方面都具有较好表现。
  - vsftpd提供的主要功能
    - 虚拟IP设置、虚拟用户、Standalone、inetd操作模式
    - 强大的单用户设置能力及带宽限流等。
  - 在安全方面
    - vsftpd从原理上修补了Wu-FTP、ProFTP、BSD-FTP等大多数FTP服务器的安全缺陷，使用安全编码技术解决了缓冲溢出问题，并能有效避免通配符类型的拒绝服务攻击。
  - 使用vsftpd作为官方网站FTP服务器的公司和团队有
    - RedHat、SuSE、Debian、GNU、GNOME、KDE、Gimp、OpenBSD等。



# 1.FTP服务器

## 1.3使用vsftpd构建FTP服务

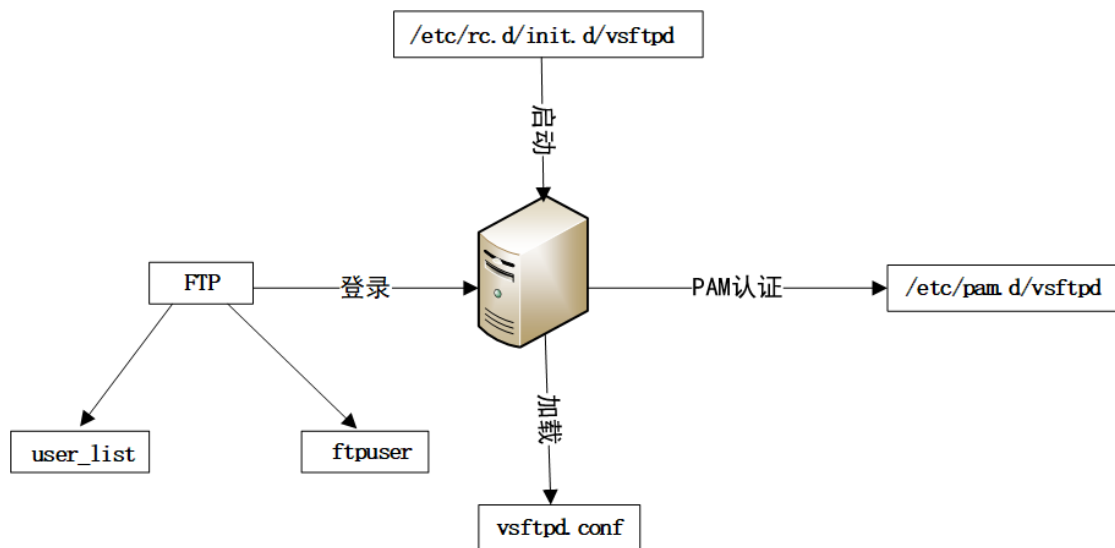
表 6-4 vsftpd 软件的常用目录文件及其说明

| 文件                          | 说明                                  |
|-----------------------------|-------------------------------------|
| /usr/sbin/vsftpd            | vsftpd 软件的主程序                       |
| /etc/vsftpd                 | vsftpd 软件的主目录                       |
| /etc/vsftpd/vsftpd.conf     | vsftpd 软件的主配置文件                     |
| /etc/pam.d/vsftpd           | 基于 PAM 的 vsftpd 软件的验证配置文件           |
| /etc/rc.d/init.d/vsftpd     | vsftpd 软件的启动脚本，也可以使用 service 进行调用启动 |
| /usr/share/doc/vsftpd-x.x.x | vsftpd 软件的文档资料路径                    |
| /usr/ftp                    | 默认的 vsftpd 软件的共享目录                  |
| /etc/vsftpd/ftpusers        | 默认的 vsftpd 软件的黑名单                   |
| /etc/vsftpd/user_list       | 修改某文件为黑名单或白名单的配置文件                  |
| /etc/logrotate.d/vsftpd     | 日志轮转备份配置文件                          |



# 1.FTP服务器

## 1.3使用vsftpd构建FTP服务



# FTP

协议

命令

软件



# 1.FTP服务器

## 1.4 任务

任务1：使用FTP命令行访问FTP资源服务

任务2：使用vsftpd建设匿名FTP服务

任务3：构建企业内部FTP文件服务



# 1.FTP服务器

## 1.4 任务

### 任务1：使用FTP命令行访问FTP资源服务

步骤1：安装ftp客户端

步骤2：使用ftp命令访问匿名FTP服务

步骤3：通过ftp命令下载资源





操作视频 / 现场演示

## ✓ 任务1：使用FTP命令行访问FTP资源服务

### ■ 任务目标：

- CentOS上使用ftp命令访问互联网匿名FTP服务
- ftp://ftp.sjtu.edu.cn
- ftp://ftp.redhat.com





## 命令指南 / 操作引导

1. [root@Project-07-Task-01 ~]# yum install -y ftp
- 2.
3. [root@Project-07-Task-01 ~]# ftp
4. ftp> open ftp.redhat.com
5. Connected to ftp.redhat.com (209.132.183.61).
6. 220 Red Hat FTP server ready. All transfers are logged. (FTP) [no EPSV]
7. Name (ftp.redhat.com:root): anonymous
8. 331 Please specify the password.
9. Password:
10. 230 Login successful.
11. Remote system type is UNIX.
12. Using binary mode to transfer files.
13. ftp> dir
14. 227 Entering Passive Mode (209,132,183,61,123,106)
15. 150 Here comes the directory listing.
16. lrwxrwxrwx 1 ftp ftp 1 Dec 19 2009 pub -> .
17. drwxr-xr-x 34 ftp ftp 4096 Apr 17 11:46 redhat
18. drwxr-xr-x 3 ftp ftp 4096 Sep 10 2019 suse
19. 226 Directory send OK.
20. ftp> cd pub
21. 250 Directory successfully changed.
22. ftp> ls



# 1.FTP服务器

## 1.4 任务

### 任务2：使用vsftpd建设匿名FTP服务

步骤1：安装vsftpd

步骤2：部署匿名FTP服务

步骤3：配置FTP服务器的安全防护措施

步骤4：通过FTP命令进行服务测试

步骤5：通过FileZilla Client测试FTP服务





操作视频 / 现场演示

## ✓ 任务2：使用vsftpd建设匿名FTP服务

### ■ 任务目标：

- 通过vsftpd建设匿名FTP服务
- 完成FTP服务器的安全配置
- 在本地主机上通过FileZilla Client测试FTP服务





## 命令指南 / 操作引导

1. [root@Project-07-Task-01 ~]# yum install -y vsftpd
2. [root@Project-07-Task-01 ~]# systemctl start vsftpd
3. [root@Project-07-Task-01 ~]# systemctl enable vsftpd
4. [root@Project-07-Task-01 ~]# systemctl status vsftpd
5. [root@Project-07-Task-01 ~]# systemctl is-enabled vsftpd
- 6.
7. [root@Project-07-Task-01 ~]# cp /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.bak
8. [root@Project-07-Task-01 ~]# vi /etc/vsftpd/vsftpd.conf
- 9.
10. [root@Project-07-Task-01 ~]# cat /etc/vsftpd/vsftpd.conf | grep ^[^#]
11. anonymous\_enable=YES
12. write\_enable=YES
13. anon\_root=/var/ftp/pub
14. anon\_world\_readable\_only=YES
15. anon\_upload\_enable=YES
16. anon\_mkdir\_write\_enable=YES
17. anon\_other\_write\_enable=YES
18. anon\_umask=077
19. dirmessage\_enable=YES
20. xferlog\_enable=YES
21. connect\_from\_port\_20=YES
22. xferlog\_std\_format=YES
23. ftpd\_banner=Welcome to linux lesson FTP service.
24. chroot\_local\_user=YES
25. allow\_writeable\_chroot=YES
26. listen=no
27. listen\_ipv6=YES
28. pam\_service\_name=vsftpd
29. userlist\_enable=YES
- 30.





## 命令指南 / 操作引导

1. [root@Project-07-Task-01 ~]# sestatus
2. [root@Project-07-Task-01 ~]# setsebool -P ftpd\_anon\_write on
3. [root@Project-07-Task-01 ~]# setsebool -P ftpd\_full\_access on
4. [root@Project-07-Task-01 ~]# getsebool -a | grep ftp
5. ftpd\_anon\_write --> on
6. ftpd\_connect\_all\_unreserved --> off
7. ftpd\_connect\_db --> off
8. ftpd\_full\_access --> on
9. ftpd\_use\_cifs --> off
10. ftpd\_use\_fusefs --> off
11. ftpd\_use\_nfs --> off
12. ftpd\_use\_passive\_mode --> off
13. httpd\_can\_connect\_ftp --> off
14. httpd\_enable\_ftp\_server --> off
15. tftp\_anon\_write --> off
16. tftp\_home\_dir --> off
- 17.
- 18.
19. [root@Project-07-Task-01 ~]# systemctl status firewalld
20. [root@Project-07-Task-01 ~]# systemctl is-enabled firewalld
21. [root@Project-07-Task-01 ~]# firewall-cmd --permanent --zone=public --add-service=ftp
22. [root@Project-07-Task-01 ~]# firewall-cmd --reload
23. [root@Project-07-Task-01 ~]# firewall-cmd --zone=public --list-all
- 24.
25. [root@Project-07-Task-01 ~]# systemctl restart vsftpd



# 1.FTP服务器

## 1.4 任务

### 任务3：构建企业内部FTP文件服务

步骤1：项目规划与部署方案

步骤2：安装vsftpd

步骤3：使用pam配置账户

步骤4：配置服务器安全措施

步骤5：测试FTP服务





操作视频 / 现场演示

### ✓ 任务3：构建企业内部FTP文件服务

#### ■ 任务目标：

- 企业内部FTP文件服务的方案规划
- 完成FTP文件服务的部署
- 完成FTP文件服务的测试



某企业为了实现文件资源的共享，需要建设FTP文件服务。

### 基本需求：

- ① 行政部、设计部、开发部有独立账号
- ② 独立账号访问，默认目录为部门目录
- ③ 所有账号能够仅能够访问本部门目录，且具有读写权限
- ④ 提供公共账号，仅允许读取公共目录的资源
- ⑤ 禁止匿名账号访问

### 解决思路：

- ① 为每个部门创建目录与账号
- ② 使用PAM进行账号管理

| 序号 | 部门  | 账号      | 权限 | 资源路径             |
|----|-----|---------|----|------------------|
| 1  | 行政部 | admin   | 读写 | /var/ftp/admin   |
| 2  | 设计部 | design  | 读写 | /var/ftp/design  |
| 3  | 开发部 | develop | 读写 | /var/ftp/develop |
| 4  | 公共  | public  | 只读 | /var/ftp/public  |





## 命令指南 / 操作引导

1. #安装vsftpd,pam
2. yum -y install vsftpd pam pam-\*
3. #配置vsftpd的服务和开机自启动
4. systemctl start vsftpd
5. systemctl enable vsftpd
6. systemctl is-enabled vsftpd
7. #创建FTP服务的存储目录
8. mkdir -p /var/ftp/admin
9. mkdir -p /var/ftp/admin/adminfolder1
10. mkdir -p /var/ftp/admin/adminfolder2
11. mkdir -p /var/ftp/design
12. mkdir -p /var/ftp/design/designfolder1
13. mkdir -p /var/ftp/design/designfolder2
14. mkdir -p /var/ftp/develop
15. mkdir -p /var/ftp/develop/developfolder1
16. mkdir -p /var/ftp/develop/developfolder2
17. mkdir -p /var/ftp/public
18. mkdir -p /var/ftp/public/publicfolder1
19. mkdir -p /var/ftp/public/publicfolder2
20. chown -R ftp:ftp /var/ftp
21. chmod -R 755 /var/ftp
22. chmod -R 777 /var/ftp/admin/adminfolder1
23. chmod -R 777 /var/ftp/admin/adminfolder2
24. chmod -R 777 /var/ftp/design/designfolder1
25. chmod -R 777 /var/ftp/design/designfolder2
26. chmod -R 777 /var/ftp/develop/developfolder1
27. chmod -R 777 /var/ftp/develop/developfolder2
28. chmod -R 777 /var/ftp/public/publicfolder1
29. chmod -R 777 /var/ftp/public/publicfolder2
30. ls -l /var/ftp/





## 命令指南 / 操作引导

1. #配置虚拟账号系统
2. cat > /etc/vsftpd/vuser\_passwd.conf < < EOF
3. admin
4. adminpwd
5. design
6. designpwd
7. develop
8. developpwd
9. public
10. publicpwd
11. EOF
12. #将文本文件的帐号及密码编译为db4的数据库文件
13. db\_load -T -t hash -f /etc/vsftpd/vuser\_passwd.conf /etc/vsftpd/vuser\_passwd.db
14. echo "虚拟账号创建完成"
15. #配置vsftpd的pam, 在文件中增加auth和account配置
16. sed -ir 's/^/#g' /etc/pam.d/vsftpd
17. echo -n '
18. auth required /lib64/security/pam\_userdb.so db=/etc/vsftpd/vuser\_passwd
19. account required /lib64/security/pam\_userdb.so db=/etc/vsftpd/vuser\_passwd
20. ' >> /etc/pam.d/vsftpd
21. echo "PAM配置完成"
22. #创建用于FTP虚拟账号服务的操作系统用户, 并禁止该用户登陆操作系统
23. userdel -rf vsftpd
24. useradd -g ftp -d /home/vsftpd -s /sbin/nologin vsftpd
25. echo "FTP服务器的操作系统账号创建完成, 账号名为vsftpd"
26. #对vsftpd的配置文件进行备份
27. mv -b /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.anon.bak
28. echo "备份vsftpd文件成功, 备份文件为/etc/vsftpd/vsftpd.conf.anon.bak"





## 命令指南 / 操作引导

1. #配置vsftpd的配置文件
2. `rm -f /etc/vsftpd/vsftpd.conf`
3. `cat > /etc/vsftpd/vsftpd.conf < <EOF`
4. #不允许匿名访问
5. `anonymous_enable=NO`
6. #设定本地用户可以访问。注意：主要是为虚拟宿主用户，如果该项目设定为NO那么所有虚拟用户将无法访问
7. `local_enable=YES`
8. #允许写操作
9. `write_enable=YES`
10. #创建或上传后文件的权限掩码
11. `local_umask=022`
12. #禁止匿名用户上传
13. `anon_upload_enable=NO`
14. #禁止匿名用户创建目录
15. `anon_mkdir_write_enable=NO`
16. #进入目录时可以显示一些设定的信息，可以通过`message_file=.message`来设置
17. `dirmmessage_enable=YES`
18. #开启日志
19. `xferlog_enable=YES`
20. #主动连接的端口号
21. `connect_from_port_20=YES`
22. #设定禁止上传文件更改宿主
23. `chown_uploads=NO`
24. #日志路径，需要对日志文件授权`chown vsftpd.vsfptd /var/log/vsftpd.log`
25. `xferlog_file=/var/log/xferlog`
26. #格式化日志
27. `xferlog_std_format=YES`
28. #禁止vsftpd账号登陆，因此写vsftpd或系统内nobody
29. `nopriv_user=vsftpd`





## 命令指南 / 操作引导

1. #设定支持异步传输功能
2. async\_abor\_enable=YES
3. #设定支持ASCII模式的上传
4. ascii\_upload\_enable=YES
5. #设定支持ASCII模式的上传
6. ascii\_download\_enable=YES
7. #登陆欢迎语
8. ftpd\_banner=Welcome to Linux Teach FTP service.
9. #限定用户在个人目录内访问。
10. chroot\_local\_user=YES
11. chroot\_list\_enable=YES
12. #限定在个人目录内访问的用户信息列表
13. chroot\_list\_file=/etc/vsftpd/chroot\_list
14. #以standalone方式启动
15. listen=YES
16. #/etc/pam.d/下的vsftpd文件
17. pam\_service\_name=vsftpd
18. #在/etc/vsftpd/user\_list中的用户将不能使用FTP
19. userlist\_enable=YES
20. #启用虚拟用户功能
21. guest\_enable=YES
22. #虚拟用户权限所对应的宿主用户，宿主用户为linux操作系统用户
23. guest\_username=vsftpd
24. #虚拟用户的vsftpd配置文件存放路径。
25. virtual\_use\_local\_privs=YES
26. #vsftpd\_config是目录，里面存放的文件名和虚拟用户名必须完全一致。
27. user\_config\_dir=/etc/vsftpd/vuser\_conf
28. EOF
29. echo "创建vsftpd的主配置文件，并完成配置"





## 命令指南 / 操作引导

1. #创建chroot\_list文件并写入文件内容
2. `rm -f /etc/vsftpd/chroot_list`
3. `touch /etc/vsftpd/chroot_list`
4. `echo vsftpd > /etc/vsftpd/chroot_list`
5. `echo "禁止FTP账号访问上级目录的配置完成"`

6. #创建虚拟用户的配置文件存放的路径
7. `rm -rf /etc/vsftpd/vuser_conf`
8. `mkdir -p /etc/vsftpd/vuser_conf`
9. `cd /etc/vsftpd/vuser_conf/`

10. #为admin用户创建vsftpd的配置文件
11. `cat > admin << EOF`
12. `local_root=/var/ftp/admin`
13. `write_enable=YES`
14. `anon_umask=022`
15. `anon_world_readable_only=NO`
16. `anon_upload_enable=YES`
17. `anon_mkdir_write_enable=YES`
18. `anon_other_write_enable=YES`
19. `EOF`

20. #为design用户创建vsftpd的配置文件
21. `cat > design << EOF`
22. `local_root=/var/ftp/design`
23. `write_enable=YES`
24. `anon_umask=022`
25. `anon_world_readable_only=NO`
26. `anon_upload_enable=YES`
27. `anon_mkdir_write_enable=YES`
28. `anon_other_write_enable=YES`
29. `EOF`





## 命令指南 / 操作引导

1. #为develop用户创建vsftpd的配置文件
2. cat > develop << EOF
3. local\_root=/var/ftp/develop
4. write\_enable=YES
5. anon\_umask=022
6. anon\_world\_readable\_only=NO
7. anon\_upload\_enable=YES
8. anon\_mkdir\_write\_enable=YES
9. anon\_other\_write\_enable=YES
10. EOF
11. #为public用户创建vsftpd的配置文件
12. cat > public << EOF
13. local\_root=/var/ftp/public
14. write\_enable=NO
15. anon\_umask=022
16. anon\_world\_readable\_only=YES
17. anon\_upload\_enable=NO
18. anon\_mkdir\_write\_enable=NO
19. anon\_other\_write\_enable=NO
20. EOF
21. echo "FTP服务账号创建并配置完成, 创建账号为admin design develop public"
22. echo "FTP服务账号的密码为【账号名】 + 【pwd】 , 例如publicpwd"





## 命令指南 / 操作引导

1. #安全性配置: SELinux Firewalld
2. systemctl is-enabled firewalld
3. firewall-cmd --permanent --zone=public --add-service=ftp
4. firewall-cmd --reload
5. echo "防火墙策略为: "
6. firewall-cmd --zone=public --list-all
7. echo "SELINUX的运行状态为: "
8. sestatus
9. setsebool -P ftpd\_anon\_write off
10. setsebool -P ftpd\_full\_access on
11. echo "SELINUX关于ftp的布尔值为: "
12. getsebool -a | grep ftp
13. echo "完成SELINUX和Firewalld的配置"
14. #重新启动vsftpd服务
15. systemctl restart vsftpd
16. echo "FTP Service is OK."



## 2.NFS服务器

### 2.1 NFS的基本原理

- NFS (Network File System) 即网络文件系统, 它由Sun公司于1985年推出的协议, 大部分的Linux发行版均支持NFS。
  - NFS允许网络中的计算机通过TCP/IP网络共享资源, 其主要功能是通过网络使不同操作系统之间可以彼此共享文件和目录。
  - NFS服务器允许NFS客户端将远端NFS服务器端的共享目录挂载到本地的NFS客户端中。
  - 在本地NFS客户端的机器看来, NFS服务器端共享的目录就如同外挂的磁盘分区和目录一样, 也就是说客户端可以透明地访问服务器中的文件系统。



## 2.NFS服务器

### 2.1 NFS的基本原理

- RPC (Remote Procedure Call Protocol) 即远程过程调用协议，属于网络文件系统的核心，也是NFS服务器工作的重要支持。
  - 由于NFS支持功能很多，例如不同文件对不同用户开放不同权限，不同的功能会启动不同的端口来传输数据等。
  - 端口不固定会造成NFS客户端与NFS服务器端的通信障碍，就需要调用RPC服务来进行规划协调。
- RPC相当于NFS客户端与NFS服务器端数据传输的桥梁。
  - RPC最主要的功能就是指定每个NFS功能所对应的端口号，并且回报给客户端，让客户端可以连接到正确的端口上进行通讯。
  - 当服务器在启动NFS时会随机选用某个端口，并主动地向RPC注册。
  - RPC则使用固定端口111来监听客户端的请求并返回客户端正确的端口，这样RPC就可以知道每个端口对应的NFS功能。



## 2.NFS服务器

### 2.1 NFS的基本原理

- NFS必须要有RPC存在时才能成功地提供服务。
  - 启动NFS之前，必须先启动RPC，否则NFS会无法向RPC注册，但是重新启动RPC时，之前注册的端口与功能的数据将会消失。
  - 重新启动RPC，需要将其管理的所有程序都重新启动，重新进行RPC注册。
  - NFS的各项功能都必须要向RPC注册，这样RPC才能了解NFS服务的各项功能的port number、PID和NFS在主机所监听的IP等，客户端才能够通过RPC的询问找到正确对应的端口。
  - NFS为RPC Server的一种。



## 2.NFS服务器

### 2.1 NFS的基本原理

- NFS服务器主要进行资源的分享，且与权限有关。
- NFS服务器启动时至少需要两个守护进程：
  - 管理客户端的登入权限
  - 管理客户端的操作权限
- NFS服务器的正常运行另外需要5个守护进程：
  - rpc.nfsd
  - rpc.mountd
  - rpc.lockd
  - rpc.statd
  - rpc.quotad



## 2.NFS服务器

### 2.2 NFS的配置文件

- NFS安装后，系统会默认创建一些目录文件。
- NFS服务所创建的重要文件如表6-9所示。

表 6-9 重要文件说明

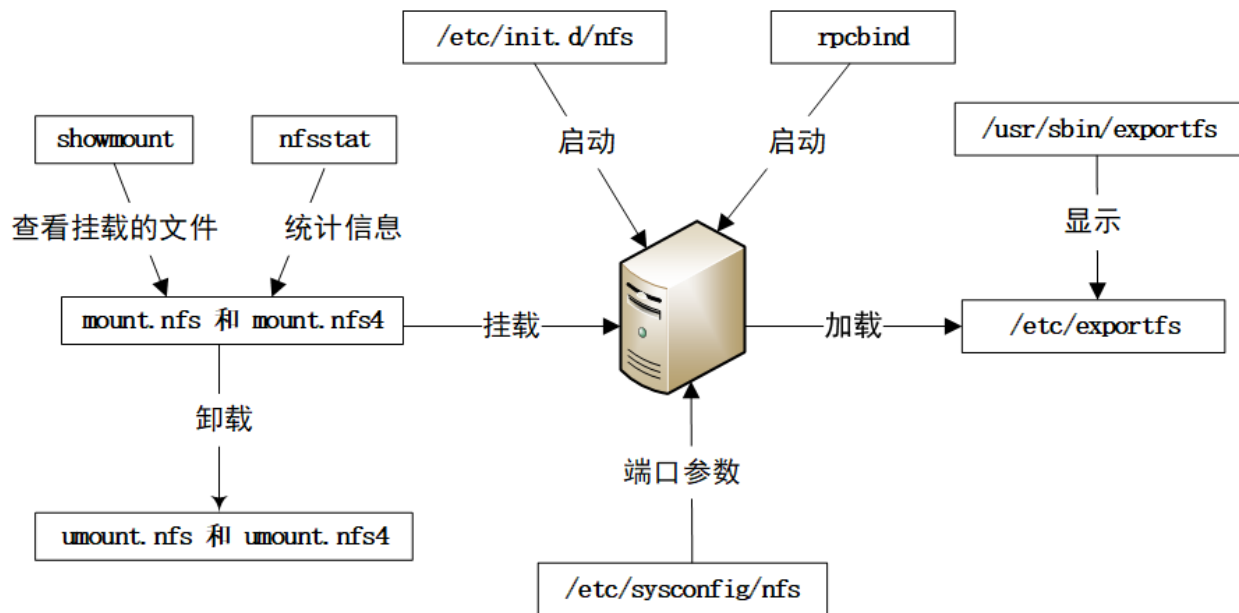
| 文件名                 | 说明                      |
|---------------------|-------------------------|
| /usr/sbin/exportfs  | 可执行文件，显示 NFS 服务中的共享文件系统 |
| /usr/sbin/showmount | 可执行文件，显示 NFS 服务器上的加载信息  |
| /usr/sbin/nfsstat   | 可执行文件，显示文件的系统状态         |
| /etc/export         | DNS 的主要配组合              |
| /etc/init.d/nfs     | 启动或停止 nfs 服务            |
| /etc/init.d/rpcbind | 启动或停止 rpcbind 服务        |
| /etc/sysconfig/nfs  | 端口参数文件                  |
| /sbin/mount.nfs     | 挂载网络文件系统                |
| /sbin/mount.nfs4    | 挂载网络文件系统（NFSv4）         |
| /sbin/unmount.nfs   | 卸载网络文件系统                |
| /sbin/unmount.nfs.4 | 卸载网络文件系统（NFSv4）         |



## 2.NFS服务器

### 2.2 NFS的配置文件

- NFS服务文件的工作流程，如图所示。



## 2.NFS服务器

### 2.3 NFS的管理工具

#### 管理NFS服务器共享的文件系统：

exportfs

##### 【功能】

exportfs 命令用于管理 NFS 服务器共享的文件系统，其配置文件为“/etc/exports”。该命令可以直接指定 NFS 的共享文件系统。

##### 【语法】

```
exportfs [选项] [参数]
```

##### 【选项说明】

exportfs 命令的选项及说明，如表 6-10 所示。

表 6-10 exportfs 命令选项说明

| 选项 | 说明                                                       |
|----|----------------------------------------------------------|
| -a | 导出或卸载所有目录                                                |
| -d | 开启调试功能                                                   |
| -o | 指定一系列导出选项(如 rw, async, root_squash)                      |
| -i | 忽略/etc/exports 和/etc/exports.d 目录下文件                     |
| -r | 重新共享所有目录                                                 |
| -u | 卸载(即不再导出)一个或多个导出目录                                       |
| -f | 如果/proc/fs/nfsd 或/proc/fs/nfs 已被挂载，即工作在新模式下，该选项将清空内核中导出表 |
| -v | 显示当前操作的详细信息                                              |
| -s | 显示适用于/etc/exports 的当前导出目录列表                              |

##### 【参数说明】

exportfs 命令的参数及说明，如表 6-11 所示。

表 6-11 exportfs 命令参数说明

| 参数     | 说明                                       |
|--------|------------------------------------------|
| 共享文件系统 | 指定要通过 NFS 服务器共享的目录，其格式为“/home/directory” |



## 2.NFS服务器

### 2.3 NFS的管理工具

#### 挂载网络文件系统： nfsstat

##### 【功能】

nfsstat 命令用于显示关于 NFS 和到内核的远程过程调用（RPC）接口的统计信息，该命令也可对这些信息进行初始化。

##### 【语法】

```
nfsstat [选项]
```

##### 【选项说明】

nfsstat 命令的选项及说明，如表 6-12 所示。

表 6-12 nfsstat 命令选项说明

| 选项     | 说明                              |
|--------|---------------------------------|
| -s     | 仅显示服务器端的状态信息。默认显示服务器端个客户端的状态    |
| -c     | 仅显示客户端的状态信息                     |
| -n     | 仅显示 NFS 状态信息，默认显示 NFS 状态信息和 RPC |
| -2/3/4 | 仅列出 NFS 版本 2/3/4 的状态            |
| -m     | 打印以加载的 nfs 文件系统状态               |
| -r     | 仅打印 rpc 状态                      |
| -o     | 显示自定义的设备信息                      |
| -l     | 以列表的形式打印信息                      |
| -s     | 打印目前的统计数据                       |



# 2.NFS服务器

## 2.3 NFS的管理工具

显示NFS服务器上的  
加载信息：  
showmount

**【功能】**  
showmount 命令用于查询“mountd”守护进程，获取远程 NFS 服务器上的 NFS 文件系统的加载信息。

**【语法】**  
showmount [选项]

**【选项说明】**  
showmount 命令的选项及说明，如表 6-13 所示。

表 6-13 showmount 命令选项说明

| 选项           | 说明                             |
|--------------|--------------------------------|
| -a           | 以 host:dir 这样的格式来显示客户主机名和挂载点目录 |
| -d           | 仅显示被客户挂载的目录名                   |
| -e           | 显示 NFS 服务器的输出清单                |
| -h           | 显示帮助信息                         |
| -v           | 显示版本信息                         |
| --no-headers | 不输出描述头部信息                      |



## 2.NFS服务器

### 2.4 任务

#### 任务4：工作组内的网络共享存储服务

步骤1：规划网络共享存储服务的方案

步骤2：部署NFS

步骤3：配置NFS服务器的安全

步骤4：在Windows上使用网络共享存储服务

步骤5：在Linux上使用网络共享存储服务





操作视频 / 现场演示

#### ✓ 任务4：工作组内的网络共享存储服务

##### ■ 任务目标：

- 规划网络共享存储服务的方案
- 部署实现网络共享存储服务
- 在Windows上访问网络共享存储服务
- 在Linux上访问网络共享存储服务



某设计工作室拥有大量的数字资源，若存储在本地会占用主机大量存储，且不利于资源共享。

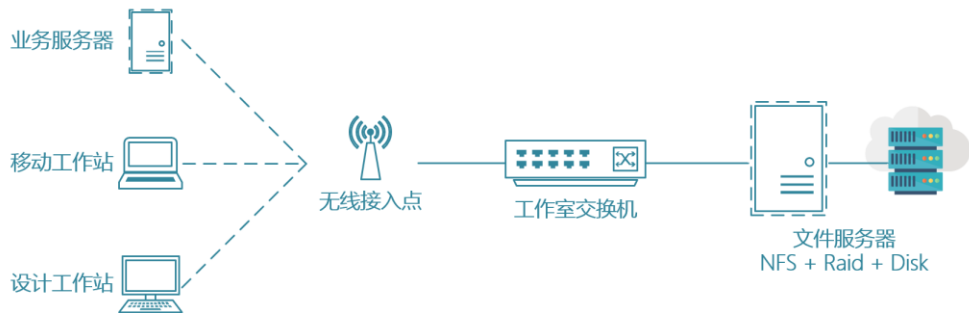
现需要构建公共网络存储，实现灵活的资源读取和共享。

### 基本需求：

- ① 建设大容量高可靠的网络共享存储服务，在存储服务器上安装大量磁盘并通过Raid技术实现存储容灾。
- ② 提供超过20TB的存储容量，实现大容量网络存储服务。
- ③ 支持MacOS、Linux、Windows等多操作系统。

### 解决思路：

- ① 通过NFS建设网络存储服务
- ② 仅允许工作室内部网络的设备访问
- ③ 支持多种操作系统进行磁盘挂载



## 部署NFS服务器





## 命令指南 / 操作引导

1. #创建用于共享的目录
2. [root@Project-07-Task-02 ~]# mkdir /opt/WorkGroupShare
3. [root@Project-07-Task-02 ~]# chmod 777 /opt/WorkGroupShare
4. #安装nfs服务
5. [root@Project-07-Task-02 ~]# yum install -y nfs-utils rpcbind
6. #对nfs服务进行配置
7. [root@Project-07-Task-02 ~]# systemctl start rpcbind
8. [root@Project-07-Task-02 ~]# systemctl enable rpcbind
9. [root@Project-07-Task-02 ~]# systemctl start nfs-server
10. [root@Project-07-Task-02 ~]# systemctl enable nfs-server
11. #对nfs进行配置确认
12. [root@Project-07-Task-02 ~]# systemctl is-enabled rpcbind
13. [root@Project-07-Task-02 ~]# systemctl status rpcbind
14. [root@Project-07-Task-02 ~]# systemctl is-enabled nfs-server
15. [root@Project-07-Task-02 ~]# systemctl status nfs-server
16. #创建nfs共享
17. [root@Project-07-Task-02 ~]# echo "/opt/WorkGroupShare 10.10.2.0/24(rw,root\_squash,no\_all\_squash,sync,insecure)" >> /etc/exports
18. [root@Project-07-Task-02 ~]# systemctl reload nfs-server
19. [root@Project-07-Task-02 ~]# exportfs -rv
20. [root@Project-07-Task-02 ~]# showmount -e





## 命令指南 / 操作引导

1. #配置nfs服务器的防火墙和SELinux
2. [root@Project-07-Task-02 ~]# rpcinfo -p
3. [root@Project-07-Task-02 ~]# firewall-cmd --permanent --zone=public --add-service=nfs
4. [root@Project-07-Task-02 ~]# firewall-cmd --permanent --add-port=111/tcp
5. [root@Project-07-Task-02 ~]# firewall-cmd --permanent --add-port=111/udp
6. [root@Project-07-Task-02 ~]# firewall-cmd --permanent --add-port=2049/tcp
7. [root@Project-07-Task-02 ~]# firewall-cmd --permanent --add-port=2049/udp
8. [root@Project-07-Task-02 ~]# firewall-cmd --permanent --add-port=20048/tcp
9. [root@Project-07-Task-02 ~]# firewall-cmd --permanent --add-port=20048/udp
10. [root@Project-07-Task-02 ~]# firewall-cmd --reload
11. [root@Project-07-Task-02 ~]# firewall-cmd --zone=public --list-all
12. [root@Project-07-Task-02 ~]# setsebool -P nfs\_export\_all\_rw on
13. [root@Project-07-Task-02 ~]# setsebool -P nfs\_export\_all\_ro on
14. [root@Project-07-Task-02 ~]# getsebool -a | grep nfs
15. #在nfs服务器上访问nfs共享，进行服务测试
16. [root@Project-07-Task-02 ~]# mkdir /mnt/sharedisk
17. [root@Project-07-Task-02 ~]# mount -t nfs 10.10.2.126:/opt/WorkGroupShare /mnt/sharedisk
18. [root@Project-07-Task-02 ~]# cd /mnt/sharedisk
19. [root@Project-07-Task-02 sharedisk]# touch a.txt
20. [root@Project-07-Task-02 sharedisk]# ls -l
21. #查看nfs共享服务的状态
22. [root@Project-07-Task-02 sharedisk]# cd ~
23. [root@Project-07-Task-02 ~]# nfsstat



## 在Linux上访问共享存储服务



## 在MacOS上访问共享存储服务



# 在Windows 10上访问共享存储服务



## 3.Samba服务器

### 3.1 Samba的工作原理

#### □ Samba的历史渊源:

- 早期类UNIX系统中可以通过NFS让所有类UNIX系统之间实现资源共享，微软为了让Windows（即当时的DOS）系统间可以实现资源共享，提出了一个不同于NFS的SMB（Server Message Block）通信协议，使得网络中的文件系统、打印机等可以实现资源共享。
- 由于微软公司没有将SMB协议公开，如果想在UNIX与Windows共享资源却很困难，基本只有通过FTP实现。



## 3.Samba服务器

### 3.1 Samba的工作原理

#### □ Samba的历史渊源:

- 1991年, 大学生Andrew Tridgwell为了解决这个障碍, 通过对数据包的分析, 编写了Samba自由软件。
- 在类UNIX系统上启用Samba服务, 即可利用SMB协议与Windows系统之间实现资源共享等相关功能。
- Samba是开放源代码的GPL自由软件, 其解决了类UNIX与Windows之间通过SMB协议进行资源共享与访问。
  - Andrew Tridgwell: [https://en.wikipedia.org/wiki/Andrew\\_Tridgell](https://en.wikipedia.org/wiki/Andrew_Tridgell)
  - Samba: <http://www.samba.org>



search samba.org:

# SAMBA

opening windows to a wider world

[Home](#)
[think Samba](#)
[get Samba](#)
[learn Samba](#)
[talk Samba](#)
[hack Samba](#)
[contact Samba](#)
[support Samba](#)

## About Samba

Samba is the standard Windows interoperability suite of programs for Linux and Unix.

Samba is [Free Software](#) licensed under the [GNU General Public License](#), the Samba project is a member of the [Software Freedom Conservancy](#).

Since 1992, Samba has provided secure, stable and fast file and print services for all clients using the SMB/CIFS protocol, such as all versions of DOS and Windows, OS/2, Linux and many others.

Samba is an important component to seamlessly integrate Linux/Unix Servers and Desktops into Active Directory environments. It can function both as a domain controller or as a regular domain member.

## Releases



**Current stable release**  
Samba 4.9.4 (gzipped)  
[Release Notes](#) · [Signature](#)

**Release History**  
[Versions & Notes](#)

**Maintenance**  
[Patches](#) · [Security Updates](#) · [GPG Key](#)

**Future**  
[Release Planning](#) · [Roadmap](#)

## Beyond Samba



**Commercial Support**  
[Global](#) · [By Country](#)

**Conferences**  
[sambaXP](#) by SerNet  
[SDC](#) by SNIA

## Donations

Nowadays, the Samba Team needs a [dollar](#) instead of pizza :-)

## Latest News



15 January 2019  
**Samba 4.10.0rc1 Available for Download**

This is the first release candidate of the upcoming Samba 4.10 release series.

The uncompressed tarball has been signed using GnuPG (ID 6F33915B6568B7EA). The source code can be [downloaded now](#). See the [release notes](#) for more info.

08 January 2019  
**Call for Papers SambaXP 2019**

From June 4th to 6th 2019, developers and users will meet again in Goettingen, Germany at the 18th international SAMBA conference, the "samba exPerience 2019".

The call for papers and early bird registration are open until February 28th 2019. Please find all necessary information at the [conference site](#).

[Further News >>](#)

## Related Sites

- [cwrap.org](#)
- [linux-cifs.samba.org](#)
- [tallic.samba.org](#)
- [tevent.samba.org](#)
- [tdb.samba.org](#)
- [ldb.samba.org](#)
- [jcifs.samba.org](#)
- [rsync.samba.org](#)
- [ccache.samba.org](#)
- [ctdb.samba.org](#)
- [ppp.samba.org](#)



### THINK SAMBA

- What Is Samba?
- Latest News
- Planet Samba
- FAQ

### GET SAMBA

- Download Info
- How To Install
- GUIs

### LEARN SAMBA

- Docs And Books
- Wiki

### TALK SAMBA

- List Subscribe
- List Archives
- IRC
- Etiquette

### HACK SAMBA

- Devel Overview
- Git Source
- Build Farm
- Bug Reports

### CONTACT SAMBA

- Samba Team
- Donations
- Contacts For...

## 3.Samba服务器

### 3.1 Samba的工作原理

- Samba是作为类UNIX系统和Windows的通信的桥梁，在设计上是让类UNIX系统加入到Windows网络中，而不是让Windows加入类UNIX网络中。
  - 在Windows 98、WindowsMe、WindowsNT操作系统中SMB服务使用137(UDP)、138(UDP)及139(TCP)端口。
  - 在Windows 2000以后版本的操作系统中使用445(TCP)端口。



## 3.Samba服务器

### 3.1 Samba的工作原理

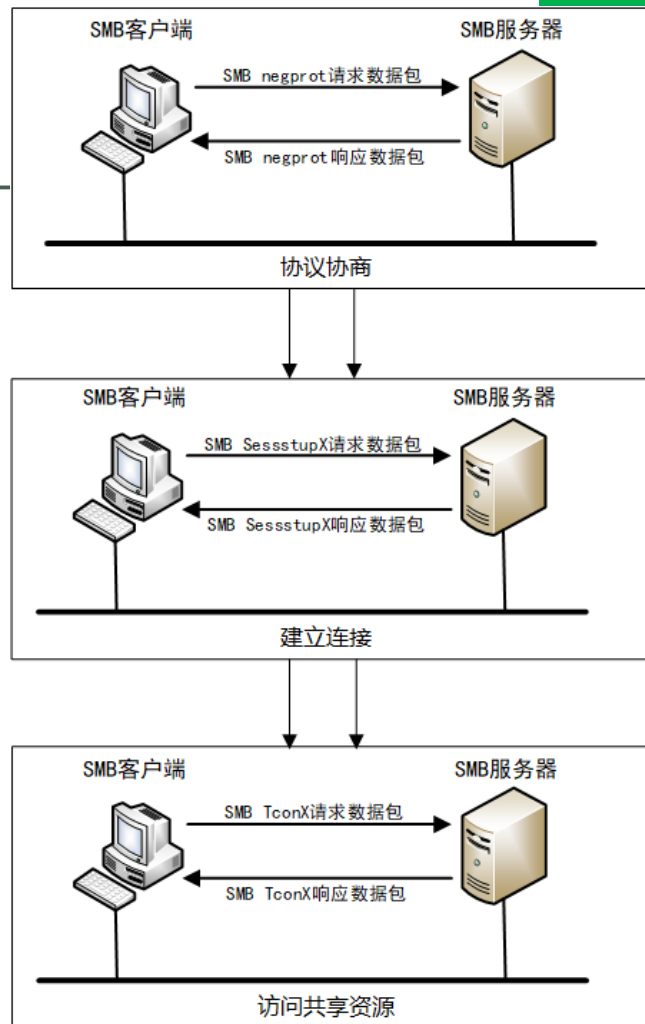
- Samba服务的两个主要进程：
  - Nmbd：进行NetBIOS名称解析，并提供浏览服务显示网络上的共享资源列表。
  - Smbd：管理Samba服务器上的共享目录、打印机等。
    - 主要是针对网络上的共享资源进行管理的服务。
    - 当访问服务器要查找共享文件时，靠smbd这个进程来管理数据传输。



## 3.Samba服务器

### □ Samba服务与Samba客户端工作流程：

- 协议协商
- 建立连接
- 访问共享资源
- 断开连接



原理

## 3.Samba服务器

### 3.1 Samba的工作原理

#### □ Samba服务器的安全模式：

- share安全级别模式
  - 客户端登录Samba服务器，不需要输入用户名和密码就可以浏览Samba服务器的资源，适用于公共的共享资源，安全性差，需要配合其他权限设置来保证安全性。
- user安全级别模式
  - 客户端登录Samba服务器，需要提交合法帐号和密码，经过服务器验证才可以访问共享资源，服务器默认为此级别模式。
- server安全级别模式
  - 客户端需要将用户名和密码，提交到指定的一台Samba服务器上验证，如果验证出现错误，客户端会用user级别访问，实现集中式的认证管理。
- domain安全级别模式
  - 加入Windows域环境中，验证工作将由Windows域控制器负责。
- ads安全级别模式
  - 具备了domain安全级别模式中所有的功能并可以具备域控制器的功能。



## 3.Samba服务器

### 3.2 Samba的配置文件

□ Samba常用的目录及文件，如表所示：

表 6-16 Samba 服务的相关文件说明

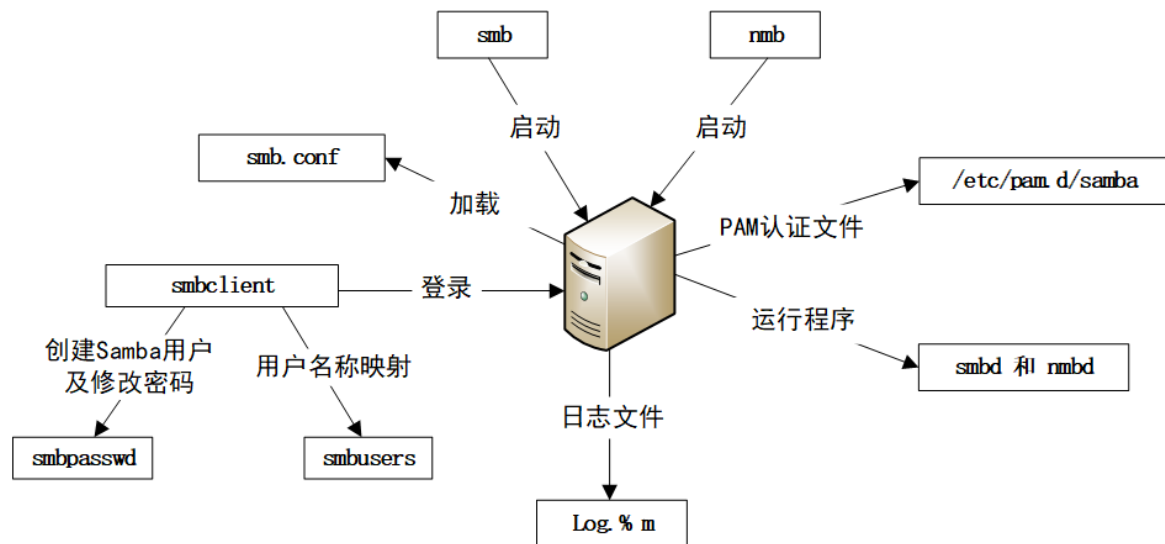
| 文件                   | 说明                       |
|----------------------|--------------------------|
| /etc/samba/smb.conf  | Samba 软件的主配置文件           |
| /etc/init.d/smb      | 控制 Samba 服务的 smbd 服务     |
| /etc/init.d/nmb      | 控制 Samba 服务的 nmbd 服务     |
| /usr/bin/smbclient   | Samba 服务的客户端工具           |
| /usr/bin/rpcclient   | 执行客户端的 MS-RPC 功能的工具      |
| /usr/bin/smbpasswd   | 修改 SMB 用户的密码             |
| /usr/bin/smbusers    | 用户名称映射文件                 |
| /usr/bin/findsmb     | 列出 SMB 名称查询上级的相关信息       |
| /usr/sbin/smbd       | 提供 SMB/CIFS 服务           |
| /usr/sbin/nmbd       | 提供 IP 命名服务上的 NetBIOS 客户端 |
| /etc/pam.d/samba     | PAM 认证文件                 |
| /usr/bin/eventlogadm | 存储 Samba 事件日志记录          |
| /usr/bin/testparm    | 检查 smb.conf 配置文件的内部正确性   |



## 3.Samba服务器

### 3.2 Samba的配置文件

□ Samba服务各文件的关系，如图所示：



## 3.Samba服务器

### 3.3 Samba使用的协议

#### □ SMB协议：

- SMB (Server Message Block) 协议是基于TCP-NETBIOS，端口使用TCP 139，TCP 445。
- SMB是微软和英特尔在1987年制定的协议，主要是作为Microsoft网络的通讯协议，用于在计算机间共享文件、打印机、串口等。
- SMB协议可以用在TCP/IP协议之上，也可以用在其它网络协议如IPX和NetBEUI之上。
- SMB一种客户端/服务器、请求/响应协议。
  - 通过SMB协议，客户端应用程序可以在各种网络环境下读、写服务器上的文件，以及对服务器程序提出服务请求。
  - 通过SMB协议，应用程序可以访问远程服务器端的文件、以及打印机、邮件槽 (mailslot)、命名管道 (namedpipe) 等资源。



## 3.Samba服务器

### 3.3 Samba使用的协议

#### □ CIFS协议：

- CIFS (Common Internet File System) 使程序可以访问远程Internet计算机上的文件并要求此计算机提供服务。
- CIFS使用客户端/服务器模式。
- CIFS在高层运行，属于应用程序协议。
  - 随着Internet的流行，Microsoft希望将SMB协议扩展到Internet上去，成为Internet上计算机之间相互共享数据的一种标准。
  - 因此将原有的几乎没有多少技术文档的SMB协议进行整理，重新命名为CIFS (Common Internet File System)，并打算将它与NetBIOS相脱离，试图使它成为Internet上的一个标准协议。



## 3.Samba服务器

### 3.4 任务

#### 任务6：构建面向全终端的文件共享服务

步骤1：规划文件共享服务的方案

步骤2：通过Samba实现文件共享服务

步骤3：配置Samba服务器的安全

步骤4：在Windows和Linux上访问samba服务

步骤5：在移动终端和智能设备上访问samba服务





操作视频 / 现场演示

## ✓ 任务6：构建面向全终端的文件共享服务

### ■ 任务目标：

- 规划文件共享服务的方案
- 部署实现文件共享服务
- 在Windows上访问文件共享服务
- 在Linux上访问文件共享服务
- 在Android上访问文件共享服务
- 在iOS上访问文件共享服务
- 在Smart TV上访问文件共享服务



某团队为了提高信息化应用水平，提高数据共享和资源服务水平，现需要构建内部网络存储，并能够全面支持移动终端等智能设备，实现灵活的资源共享。

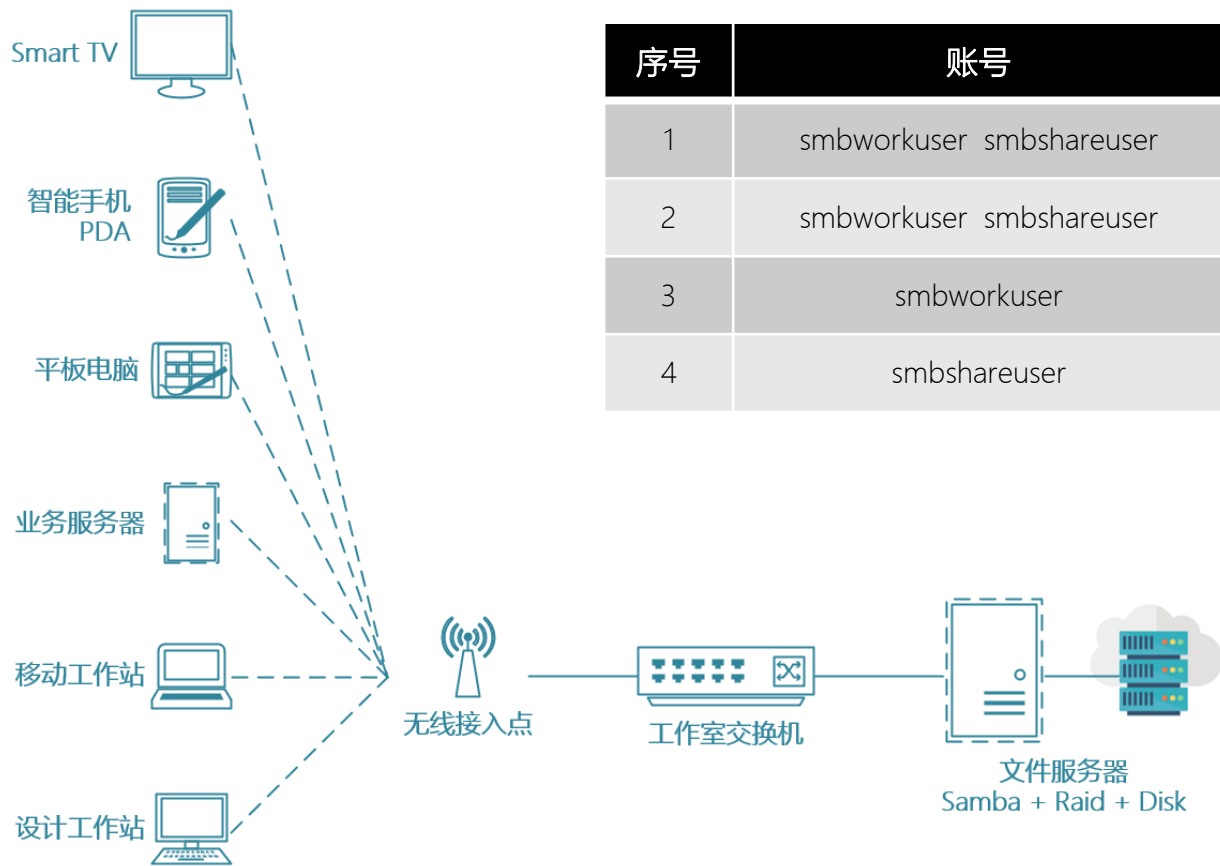
### 基本需求：

- ① 建设内部共享服务。
- ② 支持MacOS、Linux、Windows等多操作系统。
- ③ 支持智能手机、平板电脑等多样智能设备。

### 解决思路：

- ① 通过Smaba建设网络存储服务
- ② 仅允许内部网络访问
- ③ 支持多操作系统、支持多终端





| 序号 | 账号                       | 权限 | 资源路径                   |
|----|--------------------------|----|------------------------|
| 1  | smbworkuser smbshareuser | 读写 | /opt/smbfile/smbpublic |
| 2  | smbworkuser smbshareuser | 读写 | /opt/smbfile/smbshare  |
| 3  | smbworkuser              | 读写 | /opt/smbfile/smbwork   |
| 4  | smbshareuser             | 只读 | /opt/smbfile/smbwork   |



## 部署Samba服务器





## 命令指南 / 操作引导

1. [root@Project-07-Task-03 ~]# yum install -y samba samba-client
2. [root@Project-07-Task-03 ~]# mkdir -p /opt/smbfile/smbshare
3. [root@Project-07-Task-03 ~]# mkdir -p /opt/smbfile/smbwork
4. [root@Project-07-Task-03 ~]# mkdir -p /opt/smbfile/smbpublic
5. [root@Project-07-Task-03 ~]# useradd smbshareuser -s /sbin/nologin
6. [root@Project-07-Task-03 ~]# useradd smbworkuser -s /sbin/nologin
7. [root@Project-07-Task-03 ~]# smbpasswd -a smbshareuser
8. [root@Project-07-Task-03 ~]# smbpasswd -a smbworkuser
9. [root@Project-07-Task-03 ~]# chmod 777 -R /opt/smbfile/smbshare
10. [root@Project-07-Task-03 ~]# chmod 777 -R /opt/smbfile/smbwork
11. [root@Project-07-Task-03 ~]# chmod 777 -R /opt/smbfile/smbpublic
12. [root@Project-07-Task-03 ~]# ls -l /opt/smbfile/
13. [root@Project-07-Task-03 ~]# mv /etc/samba/smb.conf /etc/samba/smb.conf.bak





## 命令指南 / 操作引导

```
1. [root@Project-07-Task-03 ~]# cat > /etc/samba/smb.conf << EOF
2. [global]
3.     workgroup = hactcmitt
4.     server string = linux lesson samba server version %v
5.     netbios name = lessonsmb
6.     security = user
7.     interfaces = enp0s3
8.     hosts allow = 10.10.2.0/24
9.     max connections = 10
10.    time server = no
11.    log file = /var/log/samba/samba-log.%m
12.    max log size = 10240
13.    passdb backend = tdbsam
```

```
1. [smbpublic]
2.     comment = workgroup public share disk
3.     path = /opt/smbfile/smbpublic
4.     admin user = smbworkuser
5.     public = yes
6.     browseable = yes
7.     readonly = yes
8.     guest ok = yes
```





## 命令指南 / 操作引导

1. [smbshare]
2. comment = workgroup open share disk
3. path = /opt/smbfile/smbshare
4. admin users = smbshareuser
5. public = no
6. browseable = yes
7. valid users = smbshareuser, smbworkuser
8. readonly = no
9. read list =
10. writable = yes
11. write list = smbshareuser, smbworkuser
12. create mask = 0777
13. directory mask = 0777
14. force directory mode = 0777
15. force create mode = 07777

16. [smbwork]
17. comment = workgroup work share disk
18. path = /opt/smbfile/smbwork
19. admin users = smbworkuser
20. public = no
21. browseable = yes
22. valid users = smbshareuser, smbworkuser
23. readonly = no
24. read list = smbshareuser
25. writable = yes
26. write list = smbworkuser
27. create mask = 0777
28. directory mask = 0777
29. force directory mode = 0777
30. force create mode = 07777

31. EOF





## 命令指南 / 操作引导

1. [root@Project-07-Task-03 ~]# systemctl start smb nmb
2. [root@Project-07-Task-03 ~]# systemctl enable smb nmb
3. [root@Project-07-Task-03 ~]# systemctl status smb nmb
  
4. [root@Project-07-Task-03 ~]# firewall-cmd --permanent --zone=public --add-service=samba
5. [root@Project-07-Task-03 ~]# firewall-cmd --reload
6. [root@Project-07-Task-03 ~]# firewall-cmd --zone=public --list-all
  
7. [root@Project-07-Task-03 ~]# setsebool -P samba\_enable\_home\_dirs on
8. [root@Project-07-Task-03 ~]# setsebool -P samba\_export\_all\_ro on
9. [root@Project-07-Task-03 ~]# setsebool -P samba\_export\_all\_rw on
10. [root@Project-07-Task-03 ~]# chcon -t samba\_share\_t /opt/smbfile/smbshare
11. [root@Project-07-Task-03 ~]# chcon -t samba\_share\_t /opt/smbfile/smbwork
12. [root@Project-07-Task-03 ~]# chcon -t samba\_share\_t /opt/smbfile/smbpublic



## 在Linux上访问Samba服务



# 在Windows 10上访问Samba服务



# 在macOS上访问Samba服务



# 在Android上访问Samba服务



## 在iOS上访问Samba服务



## 在Smart TV上访问Samba服务



